

Trelby Manual

Version 2.2

Trelby developers

Copyright © 2012 Trelby developers

Table of Contents

[1. Introduction](#)

[1.1. Basic concepts](#)

[2. Basic usage](#)

[2.1. Main window](#)

[2.2. Basic editing](#)

[2.2.1. Moving around](#)

[2.2.2. Deleting elements](#)

[2.2.3. Adding elements and changing their style](#)

[2.2.4. Selecting text](#)

[2.2.5. Auto-completion](#)

[2.2.6. Miscellaneous](#)

[3. View modes](#)

[3.1. Draft](#)

[3.2. Layout](#)

[3.3. Side by side](#)

[3.4. Overviews](#)

[4. Other commands](#)

[4.1. File/New,Open,Save,Save as,Close](#)

[4.2. File/Revert](#)

[4.3. File/Print \(via PDF\)](#)

[4.4. File/Settings/\(Change|Load|Save as\)](#)

[4.5. File/Settings/Spell checker dictionary](#)

[4.6. Edit/Undo,Redo](#)

[4.7. Edit/Cut,Copy,Paste](#)

[4.8. Edit/Copy/paste, system](#)

[4.9. Edit/Select scene](#)

[4.10. Edit/Select all](#)

[4.11. Edit/Goto page](#)

[4.12. Edit/Goto scene](#)

[4.13. Edit/Insert non-breaking space](#)

[4.14. Edit/Find & Replace](#)

[4.15. Edit/Remove elements](#)

[4.16. View/Show formatting](#)

[4.17. View/Fullscreen](#)

[4.18. Script/Find next error](#)

[4.19. Script/Paginate](#)

[4.20. Scripts/Auto-completion](#)

[4.21. Script/Headers](#)

[4.22. Scripts/Locations](#)

- [4.23. Script/Title pages](#)
- [4.24. Script/Spell checker dictionary](#)
- [4.25. Script/Settings/\(Change|Load|Save as\)](#)
- [4.26. Tools/Spell checker](#)
- [4.27. Tools/Name database](#)
- [4.28. Tools/Character map](#)
- [4.29. Tools/Compare scripts](#)
- [4.30. Tools/Generate watermarked PDFs](#)
- [4.31. Help/Commands](#)
- [4.32. Help/Manual](#)
- [4.33. Help/About](#)
- [5. Settings](#)
 - [5.1. Global settings](#)
 - [5.1.1. Colors](#)
 - [5.1.2. Display](#)
 - [5.1.3. Elements](#)
 - [5.1.4. Keyboard](#)
 - [5.1.5. Misc](#)
 - [5.2. Script format settings](#)
 - [5.2.1. Elements](#)
 - [5.2.2. Formatting](#)
 - [5.2.3. Paper](#)
 - [5.2.4. PDF](#)
 - [5.2.5. PDF/Fonts](#)
 - [5.2.6. Strings](#)
 - [5.3. Script-specific settings](#)
- [6. Import and export](#)
 - [6.1. Import](#)
 - [6.1.1. ASCII Text files](#)
 - [6.1.2. Final Draft XML \(.fdx\)](#)
 - [6.1.3. Celtx \(.celtx\)](#)
 - [6.1.4. Adobe Story \(.astx\)](#)
 - [6.1.5. Fountain \(.fountain\)](#)
 - [6.1.6. Fade In Pro \(.fadein\)](#)
 - [6.2. Export](#)
- [7. Reports](#)
 - [7.1. Script report](#)
 - [7.2. Location report](#)
 - [7.3. Scene report](#)
 - [7.4. Character report](#)
 - [7.5. Dialogue chart](#)
- [8. Command line parameters](#)
- [9. Troubleshooting](#)
- [A. File format](#)

Chapter 1. Introduction

Table of Contents

[1.1. Basic concepts](#)

This is the manual for the Trelby screenwriting software. This version of the manual corresponds to

version 2.2 of the software.

1.1. Basic concepts

If you don't know anything about writing a screenplay, get a copy of a book such as David Trottier's "The Screenwriter's Bible". This manual will not cover the same ground, but focuses on concepts specific to Trelby.

Trelby divides a screenplay into a long list of elements, each with a specific *element style*. The available element styles are:

Scene

Scene header. Example: "INT. MOTEL ROOM - NIGHT".

Action

Describes action. Example: "John grabs the gun from the desk."

Character

A speaking character's name. Example: "HARRY".

Dialogue

Speech. Example: "You could do that, sure. On the other hand, if you want to live beyond the next minute or so, you might want to rethink your approach."

Parenthetical

Describes how the actor should say the following dialogue. Example: "(serious)".

Transition

Describes a non-standard transition between scenes. Example: "DISSOLVE TO:".

Shot

Describes an in-scene shot. Example: "THE NECKLACE".

Act break

Defines an act (typically used in teleplays). Example: "ACT ONE" or "TEASER".

Note

Note style is used to insert notes to yourself or to others about that part of the script. Example: "Should Bobby shoot first?".

Chapter 2. Basic usage

Table of Contents

[2.1. Main window](#)

[2.2. Basic editing](#)

[2.2.1. Moving around](#)

[2.2.2. Deleting elements](#)

[2.2.3. Adding elements and changing their style](#)

[2.2.4. Selecting text](#)

[2.2.5. Auto-completion](#)

[2.2.6. Miscellaneous](#)

This chapter discusses basic usage of Trelby.

2.1. Main window

The main window has the following parts in it in top-to-bottom order, after the standard menu bar:

Script tabs and status

Each script gets its own tab, with the tab's name being the filename of each script. You can click on each tab to change the active script to that one, or the X at the end of each tab to close that tab.

After tabs is a column showing the current element's style, a column showing what the Enter/Tab keys would do, and a column showing the current/total page number.

Main text area

This is where the actual script is shown and edited.

2.2. Basic editing

2.2.1. Moving around

The usual movement keys are available: Up/Down/Left/Right to move around, Home/End to move to start/end of line, Page Up/Page Down to move up/down one page (actually one screenful), and CTRL + Home/End to move to start/end of script.

CTRL + Up/Down moves one scene up or down.

2.2.2. Deleting elements

The Delete and Backspace keys delete text one character at a time. They also join elements to one another if pressed at, respectively, at the end or at the start of an element. The joined element will always have the style of the preceding element.

2.2.3. Adding elements and changing their style

The Enter key will always create a new element. If the cursor is at the end of an element, the new

element will be an empty one, otherwise the current element is split at the cursor position.

The Tab key will create a new element if the cursor is at the end of an element, otherwise it will switch the current element's style. Shift + Tab will always switch the element's style.

What kind of element Enter/Tab/Shift + Tab will create/switch to is dependent on the current style of the element, and is configurable by the user (see [Section 5.1.3, "Elements"](#)). These values for Enter/Tab are always shown to the right of the tab-bar.

Other ways to change an element's style are right-clicking with the mouse which will show a menu where you can change the element's style, or pressing a short-cut key for each element style. Note that if you have selected text belonging to multiple elements, all of those elements' types will be changed.

Shift + Enter or CTRL + Enter will insert a forced linebreak.

2.2.4. Selecting text

When selecting text, one end of the selection is always fixed and the other one moves around with the cursor. You can select text by clicking and dragging with the mouse (right-click unselects), or by pressing CTRL + Space which sets the fixed point of the selection at current position and then moving around, or by pressing some movement key together with Shift (e.g. Shift + Up).

You can select the current scene with CTRL + A.

After you've selected the desired text, you can use the cut/copy/paste commands to move text around, delete it, etc. You can also press Del or Backspace to simply delete it.

You can unselect the selected text by pressing Esc., or simply clicking the mouse on the screen.

2.2.5. Auto-completion

Auto-completion saves typing in scene/character/transition elements by showing a list of other elements from the script that match the text in the current element. For example, if you've written the characters "MO" in a new character element, you'll see a list of all character elements that start with "MO" and if you want, you can select one of them and that text will complete the current element.

The auto-completion list only appears after you've written at least one character. If you want to see a list of all possible choices, write a single character and delete it.

You can move around in the list with the Up/Down/Page up/Page down keys. Pressing Enter will complete the text and start a new element, while pressing End will complete the text but will not start a new element. Pressing ESC will make the list go away.

2.2.6. Miscellaneous

When inserting a new empty parenthetical element, "()" will automatically be added to it and the cursor positioned in the middle. Likewise, when switching a parenthetical element to something else, if it only contains "()", it will be replaced by "" (empty). If you write a "(" in an empty dialogue or character element it will switch to a parenthetical element. If you press Enter or Tab at

the ending ")" to start a new element, the ")" will stay in the parenthetical.

If you write "EXT." or "INT." (case doesn't matter) at the start of any element, it will switch to a scene element automatically.

Chapter 3. View modes

Table of Contents

[3.1. Draft](#)

[3.2. Layout](#)

[3.3. Side by side](#)

[3.4. Overviews](#)

This chapter describes the different view modes available. You can toggle between them using the View-menu or by keyboard shortcuts.

3.1. Draft

In 'Draft' mode, the text area consists of an empty background with the script's text drawn on top. This text includes only actual script contents, so page headers, CONTINUED/(MORE) and other automatically added texts are not shown at all. Page breaks, if enabled, are shown as a horizontal line (see [Section 5.1.2, "Display"](#)).

Reasons you might want to use this mode:

- You want to see only actual words you've written, and not be distracted by page-break related formatting.
- Page breaks do not take space, so more lines fit onto the screen.
- It's the fastest view mode.

3.2. Layout

'Layout' mode shows an almost exact representation of the page as it would appear when printed. Page borders, margins, headers, CONTINUED/(MORE) and so on are all shown at their correct places.

Ways in which the display in this mode can differ from the printed output:

- If pagination is not up-to-date, text around page breaks can sometimes show in incorrect places. We recommend that when using this mode, auto-pagination is kept on and the interval at a low value, preferably 1 second (see [Section 5.1.5, "Misc"](#)).
- If you have scene continuations enabled, they will always show as plain "CONTINUED:" on top of the page, whereas in the PDF output, they will be "CONTINUED: (n)" if needed (see [Section 5.2.2, "Formatting"](#)).

3.3. Side by side

'Side by side' mode shows multiple pages side by side, as many as fit on your screen. You need to use a high resolution display and/or small fonts in order to fit more than one page on the screen at a time. Otherwise this mode is the same as the 'Layout' mode.

Note that this mode also assumes that the entire pages fit vertically on the screen, so Page up/down commands behave somewhat differently.

3.4. Overviews

There are two overview modes, one showing each character as a single pixel, and one showing each character as 4 pixels. This is unsuitable for editing so these modes are mainly useful for getting an overview of the page layouts.

Chapter 4. Other commands

Table of Contents

- [4.1. File/New,Open,Save,Save as,Close](#)
- [4.2. File/Revert](#)
- [4.3. File/Print \(via PDF\)](#)
- [4.4. File/Settings/\(Change|Load|Save as\)](#)
- [4.5. File/Settings/Spell checker dictionary](#)
- [4.6. Edit/Undo,Redo](#)
- [4.7. Edit/Cut,Copy,Paste](#)
- [4.8. Edit/Copy/paste, system](#)
- [4.9. Edit/Select scene](#)
- [4.10. Edit/Select all](#)
- [4.11. Edit/Goto page](#)
- [4.12. Edit/Goto scene](#)
- [4.13. Edit/Insert non-breaking space](#)
- [4.14. Edit/Find & Replace](#)
- [4.15. Edit/Remove elements](#)
- [4.16. View/Show formatting](#)
- [4.17. View/Fullscreen](#)
- [4.18. Script/Find next error](#)
- [4.19. Script/Paginate](#)
- [4.20. Scripts/Auto-completion](#)
- [4.21. Script/Headers](#)
- [4.22. Scripts/Locations](#)
- [4.23. Script/Title pages](#)
- [4.24. Script/Spell checker dictionary](#)
- [4.25. Script/Settings/\(Change|Load|Save as\)](#)
- [4.26. Tools/Spell checker](#)
- [4.27. Tools/Name database](#)
- [4.28. Tools/Character map](#)
- [4.29. Tools/Compare scripts](#)
- [4.30. Tools/Generate watermarked PDFs](#)
- [4.31. Help/Commands](#)
- [4.32. Help/Manual](#)

[4.33. Help/About](#)

This chapter describes commands not discussed elsewhere.

4.1. File/New,Open,Save,Save as,Close

These do what you'd expect them to do.

4.2. File/Revert

Replaces the open script with the last version of it saved to disk.

4.3. File/Print (via PDF)

This opens up the current script in your configured PDF viewer program, from which you can print it, or just see what it looks like.

4.4. File/Settings/(Change|Load|Save as)

These allow you to edit/load/save global settings (see [Chapter 5, Settings](#)).

By default, Trelby loads settings from a file named "default.conf" on startup. See [Chapter 8, Command line parameters](#) for customizing this.

4.5. File/Settings/Spell checker dictionary

This opens a dialog allowing you to edit the global spell checker user-defined dictionary. Insert new words by entering them on separate lines, or delete existing words by deleting the lines they are on.

4.6. Edit/Undo,Redo

Every edit operation you do to the script is recorded and can be undone/redone. There is a limit to the amount of memory that Trelby will use to store undo history, and different edit operations take different amounts of memory (adding a single new character takes very little memory, whereas doing a global Find&Replace can take a lot of memory), but in normal usage you should be able to undo hundreds, or even thousands, of edit operations.

There are a few occasions when undo history is otherwise trimmed that you should be aware of:

- Applying new script-specific settings clears all undo history.
- If you do a new edit operation while you are in the middle of the undo history (i.e., you have done one or more undo operations without having done follow-up redo operations to restore all your previously made changes), all recorded undo history after the current point in it will be discarded.

4.7. Edit/Cut,Copy,Paste

Cut, copy and paste work pretty much the same way as in every other program. Note that these

commands use Trelby's internal clipboard so can't be used for copying text to/from other programs. For that, see below.

4.8. Edit/Copy/paste, system

These copy/paste text to the system wide clipboard so can be used for communicating with other programs. Note that if you copy text to the clipboard from Trelby, you must keep Trelby open until you've pasted the text to the destination program.

There's two variants of the copy command: formatted and unformatted. Which one you want to use depends on what you're trying to do. Best to try out both and see which one works better.

4.9. Edit/Select scene

Select all the text of the current scene.

4.10. Edit/Select all

Select all the text in the entire script.

4.11. Edit/Goto page

This opens a dialog allowing you to type in a page number to jump to. The Enter/Escape keys can be used as shortcuts for the OK/Cancel buttons.

4.12. Edit/Goto scene

This opens a dialog allowing you to type in a scene number to jump to. The Enter/Escape keys can be used as shortcuts for the OK/Cancel buttons.

4.13. Edit/Insert non-breaking space

This adds a non-breaking space at the current cursor position. This will ensure that the line will not break at this position when it is being typed/formatted in.

4.14. Edit/Find & Replace

This opens a dialog allowing you to search for and possibly replace text. Usually it operates on all element styles, but if you press the "More" button, you can select the element styles it operates on, which is useful when renaming a character and so on.

Pressing Enter while the cursor is in either text box is equivalent to pressing the 'Find next' button. The Esc key will close the dialog.

If you're doing partial replacing, i.e. replacing some, but not all, occurrences of 'foo' with 'bar', the efficient way to do that is to press Tab until the focus is on the 'Find next' button, and then use the 'F' (Find next) and 'R' (Replace) keys on the keyboard. Press 'F' whenever you don't want to replace that occurrence, and 'R' when you do want to replace it.

4.15. Edit/Remove elements

'Remove elements' opens a dialog allowing you to select one or more element styles to completely remove from the script.

You can use this for e.g. removing notes before sending out copies of the script, or if you want a version of the script with only the dialog, only the action, or only the scene headers, or whatever else you can come up with.

4.16. View/Show formatting

This toggles whether formatting information is shown. It consists of blue lines drawn at the right and left margins of each line, plus a red marker indicating the type of the line ending for that line. See fileformat.txt for information about the different marker types.

When using the 'Find errors' command, if you can't figure out what the error is, try switching this on to see if you have two elements where you thought you only had one, or something else that's not obvious in the normal display.

It's also useful for seeing exactly how much space you've got left on a given line.

4.17. View/Fullscreen

Changes the display to fullscreen mode, hiding all distractions. To exit fullscreen mode, either press F11 or press the icon at top-left corner.

4.18. Script/Find next error

'Find next error' searches through the script, starting from the cursor position, looking for errors. This means that to search the entire script for errors you should first position the cursor at the beginning of the script.

If it finds any errors, it positions the cursor on the line containing the error and displays a message detailing the error. You should fix the error, back up a few lines, and activate the command again to search for more errors. Alternatively, if you don't want to fix the error right now, move the cursor down one line if you want to search for more errors.

It can detect the following errors:

- An empty line, or a parenthetical element containing only ().
- A character element that is not followed by a parenthetical or a dialogue element.
- A parenthetical element that does not follow a character or a dialogue element.
- A dialogue element that does not follow a character or a parenthetical element.
- Various error conditions that can only arise from bugs in the program. This includes things like overlong lines, elements that have lines with different element types, invalid characters in the script, etc. You should never find any of these errors, but if you do, you should try to fix them before saving the script, otherwise you might not be able to load it again. You

should also notify us about the problem so we can fix it.

The messages for these all say "(BUG)" at the end so they can be identified from normal errors.

4.19. Script/Paginate

'Paginate' simply repaginates the script. You normally don't need to run this if you have automatic repagination enabled, but the command is available for use when automatic repagination is disabled or you want immediate repagination for some reason.

4.20. Scripts/Auto-completion

Here you can define items to be added to the auto-completion lists for Scene, Character and Transition elements, and optionally disable auto-completion for any one of them.

Transition auto-completes are pretty generic so by default there are a few of them here, but the Scene and Character lists are empty. The main use for them are TV shows, where you have a recurring cast of characters and locations, and it's nice to have them always available, even if they haven't yet been used in the current episode's script.

4.21. Script/Headers

'Headers' allows you to edit your script's headers. Headers are lines that are automatically added to the top of each page of the script, excluding title pages and the first page of the actual script.

Topmost in the window is a setting for how many empty lines you want inserted after your header lines.

After that is a listbox of all the strings in your headers. Select the text string you want to edit in this listbox, and use the add/delete buttons to add/delete text strings.

Below the listbox are the settings for the selected string. You can adjust the string text, style, alignment, line and positioning.

The text is first positioned according to the alignment setting, and then its horizontal starting position is adjusted by the number of characters given in 'X offset'. Note that this parameter can be negative, so for example, if you want your header string to start 2 characters before the left margin, select 'Alignment: Left' and 'X offset: -2'.

Any occurrences of '\${PAGE}' in the text string will be replaced by the page number in the actual headers.

Finally there are 'Preview' and 'Apply' buttons that you can use to preview your changes. 'Preview' opens up a sample page in your PDF viewer program, while 'Apply' applies the new headers to the open script, so you can e.g. see how the page count of the script would change.

4.22. Scripts/Locations

This dialog allows you to define what scenes are part of the same location (see [Section 7.2.](#)

[“Location report”](#)). The top listbox shows locations, separated by lines of "-----", while the bottom listbox shows scenes not part of a user-specified location. In the middle are 'Add' and 'Delete' buttons for moving scenes between the two listboxes. The bottom listbox allows you to select many scenes at once.

An example of how to use this dialog: select the scenes "INT. MOTEL ROOM - DAY", "INT. MOTEL ROOM - DAY - LATER" and "INT. MOTEL ROOM - NIGHT" in the bottom listbox, then click 'Add' which adds those three into a single location in the top listbox. Then, if you want to add more scenes to the same location, make sure one of the scenes for the location is selected in the top listbox, select the new scenes you want to add in the bottom listbox, and click 'Add'. If you want to add scenes to a new location, select one of the "-----" lines in the top listbox as the destination before clicking 'Add'.

4.23. Script/Title pages

'Title pages' allows you to edit your script's title pages. You can have as many title pages as you want, or none at all.

At the top of the window are add/delete/move/next buttons. 'Add' adds a page, 'Delete' deletes the current page, 'Move' swaps the current page with the next page, and 'Next' changes the view to the next page.

The strings listbox contains all text strings for the current page. Select the text string you want to edit in this listbox, and use the add/delete buttons to add/delete text strings.

On the right side of the listbox is a coarse preview showing the approximate location of text strings on the current page, with the selected text string shown in red.

At the bottom of the window are the settings for the selected text string. There's the text itself (which can consist of multiple lines), the position of it, and its font, size, and style.

The alignment setting has three possible options:

Left

Left-justified. X-Pos setting means the horizontal position of the leftmost edge of the first character.

Center

Centered on the page. X-Pos setting is ignored.

Right

Right-justified. X-Pos setting means the horizontal position of the rightmost edge of the last character.

Finally there's a 'Preview' button that you can use to get an accurate preview of all of the title pages.

4.24. Script/Spell checker dictionary

This opens a dialog allowing you to edit the script-specific spell checker dictionary. Insert new words by entering them on separate lines, or delete existing words by deleting the lines they are on.

4.25. Script/Settings/(Change|Load|Save as)

These allow you to edit/load/save script format settings (see [Section 5.2, “Script format settings”](#)).

4.26. Tools/Spell checker

This spell checks the current script beginning at the current position. If you want to spell check the entire script, be sure to position the cursor at the first line of the script.

If it finds any words it doesn't recognize, it jumps to that position in the script, highlights the word, and opens a dialog window that has a text box for editing the word, and the following 5 buttons:

Replace

Replace the word in the script with the (modified) one in the text box. You can also do this by pressing Enter in the text box.

Skip

Skip the word for now.

Add to script dictionary

Add the word to the script-specific dictionary.

Add to global dictionary

Add the word to the (super-)global dictionary.

Suggest replacement

Search for the 5 most similar known words and allow choosing one of them. If selected, that word is put in the text box and you can either edit it further or press 'Replace' at once to replace the word in the script.

Note that this only looks at words with the same two first characters as the misspelled word (mostly for speed reasons). So while it can correct "accommdoate" to "accommodate", it can't correct "cmoputer" to "computer".

After pressing any of the above buttons, the spell checker looks for the next misspelled word. If none are found, it closes the dialog. You can also press the Esc key at any time to close the dialog.

Note that the spell checker currently only supports the English language.

4.27. Tools/Name database

'Name database' opens a window allowing you browse and search through a database of more than 200,000 character names, each name being categorized by its nationality and sex.

On the left side of the window is a list of the name categories. The main groups available are nationalities, but there are some other interesting groups, e.g. Shakespeare, Medieval, Flowers etc.

Select the groups you want to search in, either a single one by simply clicking on it, or multiple ones by holding down CTRL and then clicking on multiple items. If you want to go back to the default of having all the groups selected, press the 'Select all' button at the bottom of the list.

On the upper right side of the window are the other search parameters. There's a text box in which you can write the text you want to search for, and two radio button groups.

The first one affects how your search string is used. The string has to match either a) at the start of the name, b) anywhere in the name or c) at the end of the name.

The second one selects whether male/female/both names are included in the search results.

The search is always case-insensitive, and a blank search string matches all names.

After having set all the search parameters, press the 'Search' button, at which point the results list will change to contain only the names matching the search parameters. A count of the matching names is shown at the bottom of the window.

You can select names from the result list and press 'Insert' to insert that name at your script at the current cursor position.

4.28. Tools/Character map

'Character map' opens a window allowing you to insert any character from the supported ISO-8859-1 (i.e. Latin-1) character set into your script.

4.29. Tools/Compare scripts

You need two different versions of the same script open to use 'Compare scripts'. Select the older one as the first script and the newer one as the second script. Trelby then goes through the scripts looking for changes between them and displays the results as a PDF file.

There's a checkbox allowing you to use the same configuration parameters for the scripts you're comparing. This is useful if the scripts differ in their formatting settings, because in that case practically the entire script would show up as changed.

The PDF file contains only the changed parts of the script. The changes are color coded: Deleted lines are red, added lines are green, non-changed lines (shown for context) are white, and marker lines for single-line changes are yellow. The changed parts are separated by two gray lines.

4.30. Tools/Generate watermarked PDFs

This tool allows you to generate multiple watermarked PDFs to distribute to multiple people.

Watermarks are handy when you want to let your readers know overtly that the script is meant only for their viewing, and is not to be distributed.

Enter details like directory, filename prefix, and watermark font size. You can have two lines of watermarks per file, and the first line is common in all files. The second line for each file is provided as a list under the "Watermarks" entry box.

The watermark in the PDF is applied as slanted text, in light grey color, as part of the background of the page. The watermark is applied on every page.

A unique ID is also saved in the 'Keywords' metadata of the PDF file, as a second source of identification of a PDF (if it is leaked with only the watermark removed).

4.31. Help/Commands

'Commands' opens a window showing all the commands and their current keyboard shortcuts. It also has a menu allowing you to save this as a HTML file if you e.g. want to print it for reference.

4.32. Help/Manual

'Manual' opens this manual you're reading right now.

4.33. Help/About

'About' opens the about screen for Trelby, showing information such as the version of the program. It also shows a random famous movie quote.

Chapter 5. Settings

Table of Contents

[5.1. Global settings](#)

[5.1.1. Colors](#)

[5.1.2. Display](#)

[5.1.3. Elements](#)

[5.1.4. Keyboard](#)

[5.1.5. Misc](#)

[5.2. Script format settings](#)

[5.2.1. Elements](#)

[5.2.2. Formatting](#)

[5.2.3. Paper](#)

[5.2.4. PDF](#)

[5.2.5. PDF/Fonts](#)

[5.2.6. Strings](#)

[5.3. Script-specific settings](#)

Trelby has 4 levels of settings: super-global settings, global settings, script format settings, and script-specific settings.

Super-global settings are the current view mode, window size and position, and the global spell

checker dictionary. There can only be one set of these.

Global settings are things like interface colors, keyboard shortcuts, display fonts and other settings that only affect the user interface of the program, not the script output. You can have multiple sets of these and switch between them at will (see [Section 4.4, “File/Settings/\(Change|Load|Save as\)”](#)). These settings are platform-dependent, so you can't share these between Windows/Linux machines, and even sharing between different machines on the same platform could be unreliable if you don't have the exact same fonts, PDF viewer applications, etc. on the machines.

Script format settings are things like paper sizes, indentation/line widths/font styles for the different element types, whether to include scene numbers, and so on that directly affect script formatting and output but are not tied to any specific script, in the sense that they can be reused for different scripts with no changes. You can easily save/load these, so you can have ready-made settings for different scenarios. This, and script-specific settings, are completely platform-neutral, guaranteeing the exact same script output whatever platform you use.

Script-specific settings are things that are specific to one particular script, like title pages, header strings, and auto-completion settings, and so can't be reused for different scripts without changes. However, if you want to have a template for these (e.g. your contact information on the title page is always the same, or you're writing several episodes of the same TV show), it's easy: just fill in the information you want, and save the script as "my_template.trel", then when you want to use that as a template, open that file, change what's needed, and save the script under a new name.

5.1. Global settings

The global settings dialog is found under File/Settings/Change. It has a list of categories on the left, the selected category's settings on the right, and three buttons, Apply, Cancel and OK on the bottom. Apply updates the settings to the ones given in the dialog so you can easily preview your changes without closing the dialog. Note that pressing Cancel after Apply has no effect, it only cancels changes that haven't yet been committed.

The following sections detail the various settings.

5.1.1. Colors

Color configuration. These only affect the display, the PDF output is always black on white.

The 'Use per-element-type colors' checkbox controls whether to use the same text color for all element types (Text foreground) or whether to use the per-element color settings (Text foreground for Action, Text foreground for Character, etc.).

5.1.2. Display

The topmost listbox allows you to configure what font you want to use for normal, bold, italic, and bold + italic text. Note that you should use the same size font for each one.

(Draft view mode only) The row spacing setting affects how many pixels there are between each row. It has an effect only in the 'Draft' view mode.

(Draft view mode only) The 'Page break lines to show' setting affects how page breaks are shown. If

you select 'None', they are not shown at all. 'Normal', you see solid gray lines in the scripts indicating actual page break positions. 'Normal + unadjusted', you see the solid lines, but you also see a dotted gray line, that shows where the page could have ended, but that the page break position was adjusted backwards to honor the settings in [Section 5.2.2, “Formatting”](#).

This is useful when you've finished your script and you're going through it trying to reduce the page count. If, for example, you have three lines, a Character, a Parenthetical, and a Dialogue element, all one line, and the unadjusted page break line is after the Parenthetical line, and the actual page break line is above the character line, you know that if you can trim just one line from the page, all three lines will then fit on the page instead of being pushed to the next page.

5.1.3. Elements

This is where you configure the interaction parameters for each element style.

At the top is a combo box where you specify what element style you're configuring. All things below that are element-specific settings.

The 'Enter/Tab creates' and 'Tab/Shift+Tab switches to' affect what element is created when Enter or Tab is pressed in this element style, or what element style Tab/Shift+Tab switches the element to. See [Section 2.2.3, “Adding elements and changing their style”](#) for more information.

5.1.4. Keyboard

Here you can configure keyboard shortcuts for each command. Note that a few command/keys (Enter, Tab, etc.) can't be changed because they're an integral part of the program's design.

5.1.5. Misc

The default script directory setting allows you to specify the default directory for File/Open dialog on program startup.

You can specify what application you want to use as your PDF viewer. On Windows, this is usually Adobe Acrobat, and the path is something like "C:\Program Files\Adobe\Acrobat 7.0\Reader\AcroRd32.exe". On Linux, you can also use Adobe Acrobat, path being something like "/usr/local/Adobe/Acrobat7.0/bin/acroread".

You can also specify arguments to give to the PDF viewer program.

'Auto-capitalize sentences', if on, auto-capitalizes sentences as you write them. If it capitalizes something you don't want capitalized, write another letter, and then replace the capitalized letter by a lowercase one.

'Auto-capitalize i -> I', if on, causes solitary 'i' characters in the middle of sentences to be changed to a 'I' character automatically.

'When opening a script, start at last saved position' causes the cursor position be restored to the point it was at when the script was saved.

'Recenter screen on scrolling' causes moving around the script using up/down keys to scroll a bit differently: when cursor is at bottom of the screen, and you press down, instead of moving the

display down just one line, it will recenter the display on the cursor.

'Typing replaces selected text' causes selected text to be removed and replaced with the character entered whenever a key is pressed and there's an active selection.

'Check script for errors before print, export or compare', if checked, causes the program to run 'Script/Find next error' for the entire script automatically before print/export/compare operations. This is to avoid printing 100 pages and then finding there's an error on one of them, fixing of which causes repagination of the entire script, and having to print all the 100 pages again.

'Show splash screen for X seconds' can be used to customize for how long to show the splash screen on startup for, or to disable showing it completely.

'Auto-paginate interval' defines how often to repaginate the script automatically. If you have a particularly slow machine, you might want to increase this number. Note that automatic repagination is only done when you modify the script, so you can't just sit back and wait for the repagination to happen, you have to keep making changes. (You can always use Script/Paginate if you want immediate repagination.)

'Lines to scroll per mouse wheel event' needs no explaining.

5.2. Script format settings

Script format settings are found in Script/Settings/Change. It has the same Apply/Cancel/OK buttons as the global settings dialog.

5.2.1. Elements

This is where you configure the formatting parameters for each element style.

At the top is a combo box where you specify what element style you're configuring. All things below that are element-specific settings.

The Screen/Print areas hold four checkboxes, Caps/Bold/Italic/Underlined, affecting the text style for the element. Screen affects the editing screen, while Print affects PDF output.

The 'Empty lines / 10 before' setting is linespacing before the element, divided by 10. So a value of 10 means normal linespacing, 15 means 1.5 linespacing, 20 is double linespacing, etc.

The 'Empty lines / 10 between' is the same, but for intra-element linespacing, i.e. consecutive lines of the same element have this much space between them.

The Indent and Width settings specify in characters how much the element is indented (i.e. set off from the left margin) and how wide it is. 10 characters (in the default 12-point font) equal one inch.

5.2.2. Formatting

Here you can configure parameters affecting the formatting of the script.

The first two settings affect how many action and dialogue lines there must be left at the end of a page to allow breaking at that position. This only affects elements that are broken in the middle, i.e. if you have a one-line element, the page can always end after that.

The font size specifies the font size to use. 12 is the default, and you should only deviate from this if you have a very good reason.

'Include scene continueds', if enabled, inserts "(CONTINUED)" at the end of each page that has a scene that continues to the next page, and "CONTINUED: (n)" (n = number of CONTINUED pages for current scene, omitted if n = 1) for each page that continues a scene from previous page(s).

'Scene continueds indent' is the number of characters to indent the "(CONTINUED)" text with.

'Include scene numbers' causes scene numbers to be inserted to the left and right of each scene element. This is usually only done for scripts in production.

'Show line numbers' prefixes each line with its line number (starting from 1 at the top of the page). It can be considered a debug option as it has no use in normal operation, but it can be useful if you're trying to figure out printing related problems.

5.2.3. Paper

Here you can configure your output paper size and margins. You can use A4, Letter, or a custom paper size. At the bottom of the page is shown the number of lines that will fit on a page with the given settings.

5.2.4. PDF

Here you can configure parameters affecting the generation of the PDF file for printing or exporting.

'Add table of contents', if on, will cause the PDF file to have a table of contents listing all the scenes. Note that this is not printed, it's just shown in PDF viewer applications allowing one to quickly jump to the selected scene.

'Show table of contents on PDF open', if on, will cause the table of contents to be shown by default when opening the PDF file.

'Open PDF on current page', if on, will insert a command into the PDF file telling the PDF viewer application to show the current page when opening the file. Not all PDF viewer applications support this command. Note that this only affects PDF files generated through the File/Print command, File/Export command isn't affected by this. If you really want to generate a permanent PDF file opening on a non-standard page, use File/Print and save the PDF file from the PDF viewer application to a different name.

'Omit Note elements', if on, will cause all Note elements to be omitted from the PDF output.

'Draw rectangles around Note elements', if on, will cause rounded rectangles to be drawn around Note elements in the PDF output to make them stand out more.

'Show margins' draws a rectangle on each page showing the margin settings. It can be considered a debug option as it has no use in normal operation, but it can be useful if you're trying to figure out printing related problems.

5.2.5. PDF/Fonts

Here you can configure the fonts used in generating the PDF files. Normally they use the standard PDF Courier fonts that are guaranteed to work everywhere, and it is strongly recommended to leave things as they are unless you have specific reasons to use other fonts.

So, assuming you have a good enough reason, you can use the TrueType fonts of your choosing. There are four different font settings, one for each font style (Normal / Bold / Italic / Bold-Italic).

Each font setting has two parts: a 'Name' field and a 'File' field. The 'Name' field is the Postscript name of the font and is used by the PDF viewer application to recognize the font to use. Some examples of the names that would go here are 'AndaleMono', 'CourierNewPSMT', and 'BitstreamVeraSansMono-Roman'. These names are, in general, impossible for human beings to know for arbitrary fonts. See the next paragraphs for ways around this.

The 'File' field is the filename of the TrueType font to use. If you fill in this field, the font file is embedded in the generated PDF files. If you leave this field empty but fill in the 'Name' field, the font file is not embedded in the PDF file, just the 'Name' reference is used to indicate what font the PDF viewer application should use.

There are many factors affecting whether you should embed a font or not. Embedding a font makes it much more likely that the PDF file will display properly on other people's computers and/or print properly. However, it will also increase the size of the PDF files drastically, and not all fonts allow embedding in their license terms.

When you use the 'Browse' button to locate a font file, the following things happen:

- The file is checked to be a valid TrueType font file.
- The font's Postscript name is located from the font file and put in the 'Name' field.
- The licensing terms of the font are checked and if embedding is not allowed, a warning is printed.

Note that the chosen fonts have to be able to be substituted in place of the normal Courier fonts with no changes to the text layout, i.e., they have to be fixed width fonts with the same size as the Courier fonts are. If this is not the case, you will see characters overlapping each other and other errors in the output.

Also note that random fonts downloaded from the Internet are very often broken in every sense of the word. They have names that violate the Postscript/PDF specifications, they lack essential characters (it is very rare for them to contain anything outside of the ASCII character set), and their quality is bad. So, if you feel you must use something besides the normal Courier fonts, at least use professional fonts made by people who know what they're doing.

Note for Windows users: Microsoft, in their infinite wisdom, have found yet another way to obstruct people from using their computers. When you use the 'Browse' button and navigate to e.g. the 'C:\Windows\Fonts' directory to choose a TrueType font file, the files are listed correctly but you can't select them! One wonders what goes on in those meetings at Microsoft where things like this are decided... "Why would anyone ever want to select a font file from a normal file open dialog?" "Gee Bob, I don't know, maybe we should disallow that so that users aren't confused?" "Good idea

Jim, let's do that!").

To work around this bit of brokenness, once you're in the directory containing the font file you want to use, switch the keyboard focus to the filename entry field and type in the first few characters of the filename, after which a completion pop-up should show up from which you can select the correct filename.

5.2.6. Strings

Here you can customize the strings that are automatically added to the script in certain situations:

(CONTINUED)

Added at the end of a page when a scene spans multiple pages.

CONTINUED:

Added at the start of a page when a scene spans multiple pages.

(MORE)

Added at the end of a dialogue element that spans multiple pages.

(cont'd)

Added after the speaker's name at the start of a page when a dialogue element spans multiple pages.

5.3. Script-specific settings

Descriptions for these start at [Section 4.20, "Scripts/Auto-completion"](#).

Chapter 6. Import and export

Table of Contents

[6.1. Import](#)

[6.1.1. ASCII Text files](#)

[6.1.2. Final Draft XML \(.fdx\)](#)

[6.1.3. Celtx \(.celtx\)](#)

[6.1.4. Adobe Story \(.astx\)](#)

[6.1.5. Fountain \(.fountain\)](#)

[6.1.6. Fade In Pro \(.fadein\)](#)

[6.2. Export](#)

This chapter discusses the import/export support of the program.

6.1. Import

Existing screenplays can currently be imported from three file formats: as formatted ASCII text, Final Draft XML (.fdx), Celtx (.celtx), Adobe Story (.astx), Fountain (.fountain) and Fade In Pro

(.fadein). The following section documents the format these files should have for optimal import processing.

6.1.1. ASCII Text files

ASCII text files should have the following characteristics:

- Each element style should begin at a unique column. An exception is that Scene and Action can begin at the same column.
- There should only be one such starting column per element style, i.e. all dialogue should begin at the same column, it shouldn't sometimes start at column 25, sometimes at 27, etc.
- No production markings, e.g. scene numbers, page numbers, cast lists, or anything like that.
- Elements should be separated by a single empty line.

Below is a short example of the kind of format best recognized by Trelby.

EXT. OCEAN - NIGHT

The package hits the water and is submerged. A few seconds later, a rapidly inflating liferaft appears.

The people land into the water in their parachutes. They dispose of them and swim toward the liferaft.

INT. LIFERAFT - NIGHT

They're all aboard, taking off their clothes. When they're all naked, Sarah gathers the wet clothes, throws them out, and closes the entrance to the tent-like liferaft. The men openly admire Sarah's body, as she opens a storage locker and takes out dry clothes for everyone which they then put on.

KEVIN

I know a great way to warm up.

SARAH

Sorry, I got over the "older men" thing a while back.

KEVIN

Anytime you have a flashback, let me know.

Something starts beeping.

6.1.2. Final Draft XML (.fdx)

Final Draft files saved with version 8 of the program are supported with these contents imported:

- Text of the the screenplay elements (Action, Dialog, etc) that Trelby understands.

Script notes in Final Draft are imported as notes.

Note that no formatting settings are imported at all.

6.1.3. Celtx (.celtx)

These files are created by the Celtx program.

- Screenplay elements (Action, Dialog, etc) that Trelby understands are imported. Others are converted to 'Action' type.

Note that no formatting settings are imported at all.

6.1.4. Adobe Story (.astx)

These files are created by the Adobe Story editor.

- Screenplay elements (Action, Dialog, etc) that Trelby understands are imported. Others are converted to 'Action' type.

Note that no formatting settings are imported at all.

6.1.5. Fountain (.fountain)

Fountain is an open, plaintext screenplay markup language. It is documented at <http://fountain.io/syntax>.

Almost all of the spec is supported. The below are unsupported:

- Emphasis markup (bold, italic, etc) is not supported. When importing, you can choose to retain the syntax, or remove it.
- Dual dialogue is imported as a couple of dialogues instead.
- Section/Synopsis are imported as notes.
- Comments that are part of another element are not recognized. (Standalone comments are imported as notes).
- Title pages are imported as action lines. You can also choose not to do this when importing.
- Page breaks are not supported (imported as an action line).

Note that no formatting settings are imported at all.

6.1.6. Fade In Pro (.fadein)

These files are created by the Fade In Pro screenwriting program.

- Screenplay elements (Action, Dialog, Notes, etc) that Trelby understands are imported. Others are converted to 'Action' type.

Note that no formatting settings are imported at all.

6.2. Export

You can export the script in PDF, RTF, formatted ASCII, HTML, Final Draft XML (.fdx) and Fountain (.fountain) format.

All are found in the File/Export dialog, which allows you to choose which one to export as. If you

export as a text file, Trelby will ask whether you want the page markers included or not. If you're planning on importing the script to some other program, you shouldn't include them, otherwise you probably should.

The RTF file produced uses stylesheets, which not all RTF reading programs support, so if you get unexpected output check with something like OpenOffice to see whether the problem is with the file or with the program you tried to use.

Chapter 7. Reports

Table of Contents

[7.1. Script report](#)

[7.2. Location report](#)

[7.3. Scene report](#)

[7.4. Character report](#)

[7.5. Dialogue chart](#)

This chapter discusses the various report types Trelby can generate. All the reports are in PDF format so that you can easily view them on your computer, and if necessary also get high-quality printed output.

7.1. Script report

The script report is a short, one-page report on general script statistics. It has the following information:

- Total line count of the script and line counts for each element type, both in absolute terms and as a percentage of the whole.
- Percentage of lines found in interior and exterior scenes (it determines this based solely on whether the scene element starts with INT. or EXT.)
- Maximum and average scene lengths in lines.
- Maximum and average action element lengths in lines.
- Number of different speaking characters.

7.2. Location report

The location report is similar to the scene report, but combines scenes taking place at the same location into one entity. If you always use the same scene line to describe a single location, you don't need to do anything special to get a correct location report. If, however, you have e.g. scenes named "INT. MOTEL ROOM - DAY", "INT. MOTEL ROOM - DAY - LATER" and "INT. MOTEL ROOM - NIGHT" and you'd like them to be treated as the same location, see [Section 4.22, "Scripts/Locations"](#) for details on how to do that.

The location report lists the following information for each location: list of scenes for the location (if more than one, each is followed by a percentage showing its part of the total lines of the

location), total lines in location, percentage of action lines in location, percentage of the location's total lines of the script's total lines, how many scenes the location has, count of pages that the location appears in, list of pages that the location appears in, and optionally a list of all speaking characters in the location along with the number of lines they speak.

7.3. Scene report

The scene report lists the following information for each scene: scene number, scene name, total lines in scene, percentage of action lines in scene, count of pages that the scene appears in, list of pages that the scene appears in, and optionally a list of all speaking characters in the scene along with the number of lines they speak.

7.4. Character report

'Character report' first opens a window allowing you to specify what to include in the generated report. You can select what characters are included, and what kind of information to include for each character:

Basic information

Total number of speeches, total number of lines, the percentage of all the dialogue lines in the script that this character speaks (i.e. 25% would mean this character has one-fourth of the total dialogue in the script), average number of lines per speech, total words, average number of words per speech and average number of characters per word.

Page list

Number of pages this character speaks on, followed by a list of all of those pages.

Location list

A list of all the locations this character speaks in, preceded by the number of lines spoken at each location.

7.5. Dialogue chart

'Dialogue chart' first opens a window allowing you to specify what to include in the generated chart. The 'Sorted by:' options select what orderings to include, while the 'Characters < 10 lines' option, if unselected, leaves out all characters that have less than 10 dialogue lines, which is useful for trimming out minor characters from long scripts.

It then generates a chart with one page for each selected ordering and shows that.

The chart has two axes: The horizontal one, with page numbers increasing from left to right, and the vertical one, with characters ordered by the current page's sort criterion from top to bottom.

If a character speaks on a given page, he will have a black bar on that page's position on the chart. The bar's height is scaled according to how many dialogue lines the character has on that page, allowing easy visual identification of long monologues, etc.

Every other line has a light gray background to allow easier visual matching of distant markings to character names.

You can use this chart in many ways. For example, you can at a glance see if one of your side characters disappears for 50 pages in the middle of the script, whether the villain is introduced too late, and so on.

At the top is another graph showing the percentage of action, dialogue, character and "other" lines on each page. This graph is most useful for seeing how the percentage of action lines fluctuates, allowing one to easily see where the major action scenes are, etc.

Chapter 8. Command line parameters

Trelby supports the following command line parameters:

`--conf FILENAME`

Read global settings from the given file instead of "default.conf".

`FILENAME1 FILENAME2 ...`

Open the given script files.

Chapter 9. Troubleshooting

This chapter discusses some common problems you might encounter.

- The program usually works perfectly, but sometimes it's so much slower that it's almost unusable.
 - This happens on certain machines with limited resources. Try stopping all other programs and restarting Trelby and see if that helps; if so, make it a habit of starting Trelby first so it can allocate the resources (video memory, mainly) that it needs for fast operation before starting other programs.

Appendix A. File format

The file format used by Trelby for scripts is open and documented, and third parties are encouraged to add support for the format in their programs and/or write utility programs that use the file format in some way.

For more information, see the file 'fileformat.txt' in the program's installation directory, and read through the sample script file provided. If you still have questions about the file format, feel free to contact us by email.